

New Application planning

-Molemo Khethwa MN25030017735

1. Define Functional & Non-Functional Requirements.

Types	Details
Functional	Custom user model, role-based access (Reader, Journalist, Editor), full CRUD for Articles & Newsletters, approval flow, article delivery via email/X (Twitter), REST API endpoints
Non-functional	MariaDB integration, clean file structure, DRY principles, reusable code, RESTful API with token auth, user feedback (UX), unit tests

2. PSEUDOCODE

User Authentication and Role-Based Access

When a user logs in with valid credentials:

- If the user is a **reader**, they are redirected to the homepage where they can browse articles.
- If the user is a **journalist**, they are redirected to their dashboard to manage articles.
- If the user is an **editor**, they are redirected to the moderation dashboard to review submissions.

If login fails, an error message is shown.

User Registration (Reader Only)

When a new user tries to register:

- The system ensures only the **reader** role is selectable.
- The password must meet complexity requirements (length, uniqueness, etc.).
- If valid, the user is saved and shown a success message.
- If invalid (e.g., password too common), validation errors are displayed.

Editors and journalists cannot register via the form; they are added via the Django admin interface.

Article Creation by Journalists

A logged-in journalist can access a form to write a new article.

- The journalist enters the title, content, and optionally selects a publisher.
- Upon submission, the article is saved with status set to "**submitted**".
- The article is not visible to the public until reviewed and approved.

Article Editing (Rejected Articles Only)

When a journalist views their list of articles:

- If an article has been **rejected**, an "Edit & Resubmit" button appears.
- The journalist can edit the article's content.
- On resubmission, the article status is reset to "**submitted**", and any previous rejection message is cleared.

Article Moderation by Editors

Editors see all submitted articles.

- Each article has options to **approve** or **reject**.
- When approving, the article status is changed to "**approved**", and a timestamp is saved.
- When rejecting, the editor must provide a **rejection reason**, which is saved with the article for the journalist to review.

Public and Private Article Visibility

- **Readers** can view only articles that are marked as **approved**.
- **Unauthenticated users** can also browse all approved articles, but cannot comment, subscribe, or download.
- **Journalists** can see and manage only their own articles.
- **Editors** have full access to all articles, regardless of status.

Subscriptions

Readers can subscribe to:

- Specific **journalists** to follow their published articles.
- Specific **publishers** to get articles posted under their brand.

On the subscriptions page, subscribed entities are listed with "**Unsubscribe**" buttons.

Homepage Logic

- If the user is a **reader**, they see approved articles from their subscriptions (journalists + publishers), and recommendation cards based on tags.
- If the user is unauthenticated, they see all approved articles.
- Editors and journalists see their relevant dashboard content.

Article Search and Tag-Based Recommendations

- Users can search articles by title, content, or tags.
- Tag frequency is tracked across the platform.
- Readers receive **article suggestions** based on the most common tags from previously read or subscribed content.

Exporting Articles

Logged-in users can export any single article to:

- **CSV format:** contains title, author, status, and approval date.
- **PDF format:** well-formatted printable version of the article.

Export links are available on article detail pages.

Soft Deletion

Articles are never permanently deleted.

- When deleted (moderation), the `is_deleted` flag is set to `True`.
- Soft-deleted articles are hidden from normal queries but can be restored if needed.

RESTful API (DRF Support)

The backend supports token-authenticated API endpoints:

- Create article (journalist only)
- List public articles
- List journalist or publisher-specific articles
- Approve or reject articles (editor only)
- Obtain auth token for API usage

These APIs mirror the web functionality and are secured by role-based permissions.

Testing & Coverage

You implemented:

- API tests for article creation, rejection, and duplication prevention.
- Unit tests for model defaults, article visibility, and tag processing.
- All test files are split into API and logic/unit test categories.

3.CRUD

Entity	Action	Role	View/Method	URL / Endpoint	Notes
User (Reader)	Create	Public	user_register	/register/	Readers only, via public site
	Read	All	home, article_list, article_detail	/, /articles/, /articles/<id>/	Readers see approved articles
	Update	-	-	-	Readers don't update their account
	Delete	Admin	Django admin	/admin/	Managed internally